Using Web Information for Creating Publication Venue Authority Files

Denilson Alves Pereira [1]

Berthier Ribeiro-Neto [1 2]

Nivio Ziviani [1]

Alberto H. F. Laender [1]

[1] Department of Computer Science, Federal University of Minas Gerais

[2] Google Engineering

Belo Horizonte, Brazil

The Problem

- Citations to publication venues can be written in several different forms
  Example: references to the VLDB conference
  - Proc. VLDB
  - Proceeding of the 18th VLDB Conf rence, Aug
  - Proc. of the Int. Conf. on Very Large Data Bases (VLDB)
  - International Conference on Very Large Databases
- Drawbacks
  - misspellings, spelling variants, and abbreviated forms
  - difficulty for searching and retrieval

The Problem

- **Authority files** maintain, for any given bibliographic item, a list of variant strings used as a reference to it
- CiteSeer and Google Scholar do not maintain authority files and, therefore, include a large number of bibliographic records that are frequently inconsistent
- The creation of authority files is not a simple task, since methods based on string matching tend to suffer from small recall

Our Proposal

- To use information available on the Web to create publication venue authority files
- Main steps
  - take a publication venue reference and submit it as a query to a search engine
  - parse the answers looking for variant reference forms to the same venue in the text snippets
  - use these variant forms to generate an authority file

Our Proposal

- In our authority file, each entry contains
  - a publication venue canonical name
  - an acronym (if it exists)
  - a venue type (i.e., journal, conference or workshop)
  - a mapping to various forms of writing the venue name in bibliographic citations
- Our method might select, as canonical name of a venue, a string that was not provided as input

Related Work

- L. Auld: concept of authority control
- VIAF project
- French, Powell and Schulman: experimented several techniques for creating an authority file of affiliations for authors
- D. Lee: evolving metadata, updating, and searching
- Several methods have been proposed for disambiguating author names
- Several works use the Web as a source of additional information

Our Baseline

French, Powell and Schulman (2000)

- Used approximate string matching techniques
- Created an authority file of affiliations for researchers in the Astrophysics Data System
- Their techniques can be also applied to publication venue authority files

FPS Authority Files

FPS Algorithm
input: $S = \{s_1, s_2, ..., s_n\}$
output: $C = \{(c_1, n_1), (c_2, n_2), ..., (c_p, n_p)\}$
begin
  $C \leftarrow$ clustering (S);
  for each $c_i \in C$
    $n_i \leftarrow$ most-frequent-affiliation-string (c_i);
end

Comments on FPS Method

- The “clustering" procedure they used is based on a function $d(s_i, s_j)$ that computes the distance between distinct strings s_i and s_j
- They tried different functions
  - edit-distance
  - edit-distance combined with the Jaccard similarity coefficient

Our Web-based Method

In our authority file, each entry represents a publication venue and includes

- its canonical name
- its acronym
- its type (i.e., journal, conference or workshop)
- a list of variant (publication venue) strings that refer to it in bibliographic citations

A Sample of Our Authority File

p_1: $Cluster_1$

n_1: ACM/IEEE-CS Joint Conference on Digital Libraries

a_1: JCDL

t_1: conference

l_1: “Proc. JCDL", “Joint Conference on Digital Libraries (JCDL 2003), Houston, TX, May", “Digital Libraries, 2004. Proceedings of the 2004 Joint ACM/IEEE Conference on"

p_2: $Cluster_2$

n_2: IEEE/ACM Transactions on Networking

a_2: TON

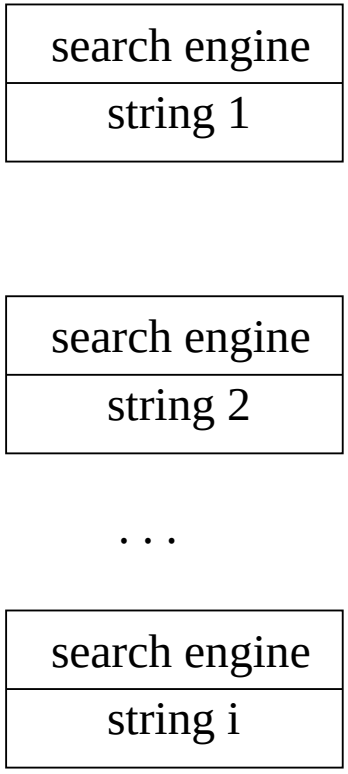
t_2: journal

l_2: “IEEE Transactions on Networking (December 1997)", “ACM Transactions on Networking", “IEEE/ACM Trans. Netw"

Summary of The Main Steps

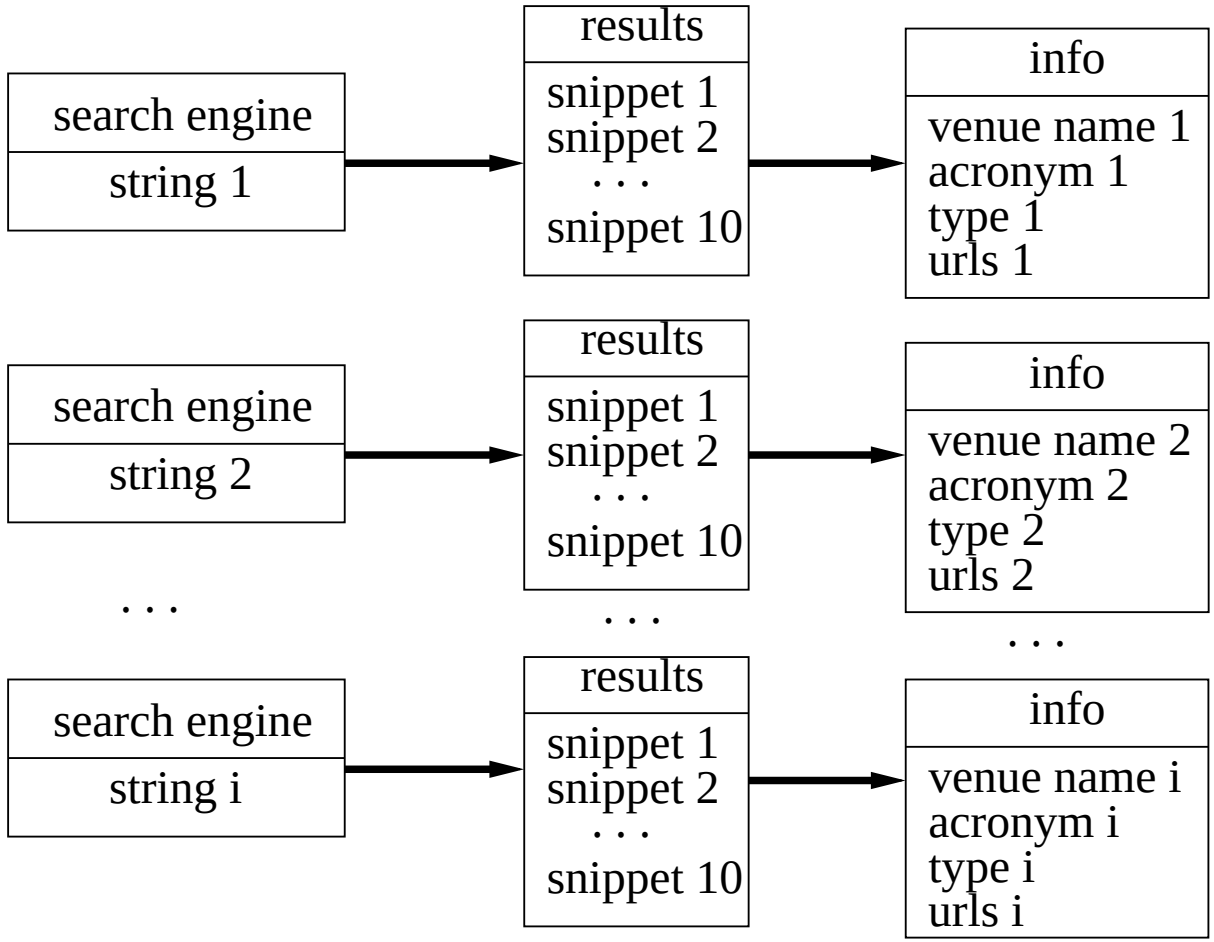
- Step 1 - Querying the search engine
  - Take as input a set of publication venue strings, normalize them, and submit each string as a query to a search engine
- Step 2 - Extracting information from the snippets
  - From each snippet, extract a publication venue name, an acronym, a type, and its URL
  - From the set of snippets of a query, select a publication venue name, an acronym, a type, and its set of URLs
- Step 3 - Clustering the data
  - Cluster the information selected in Step 2
  - Generate canonical data

Scheme for Creating PVAF

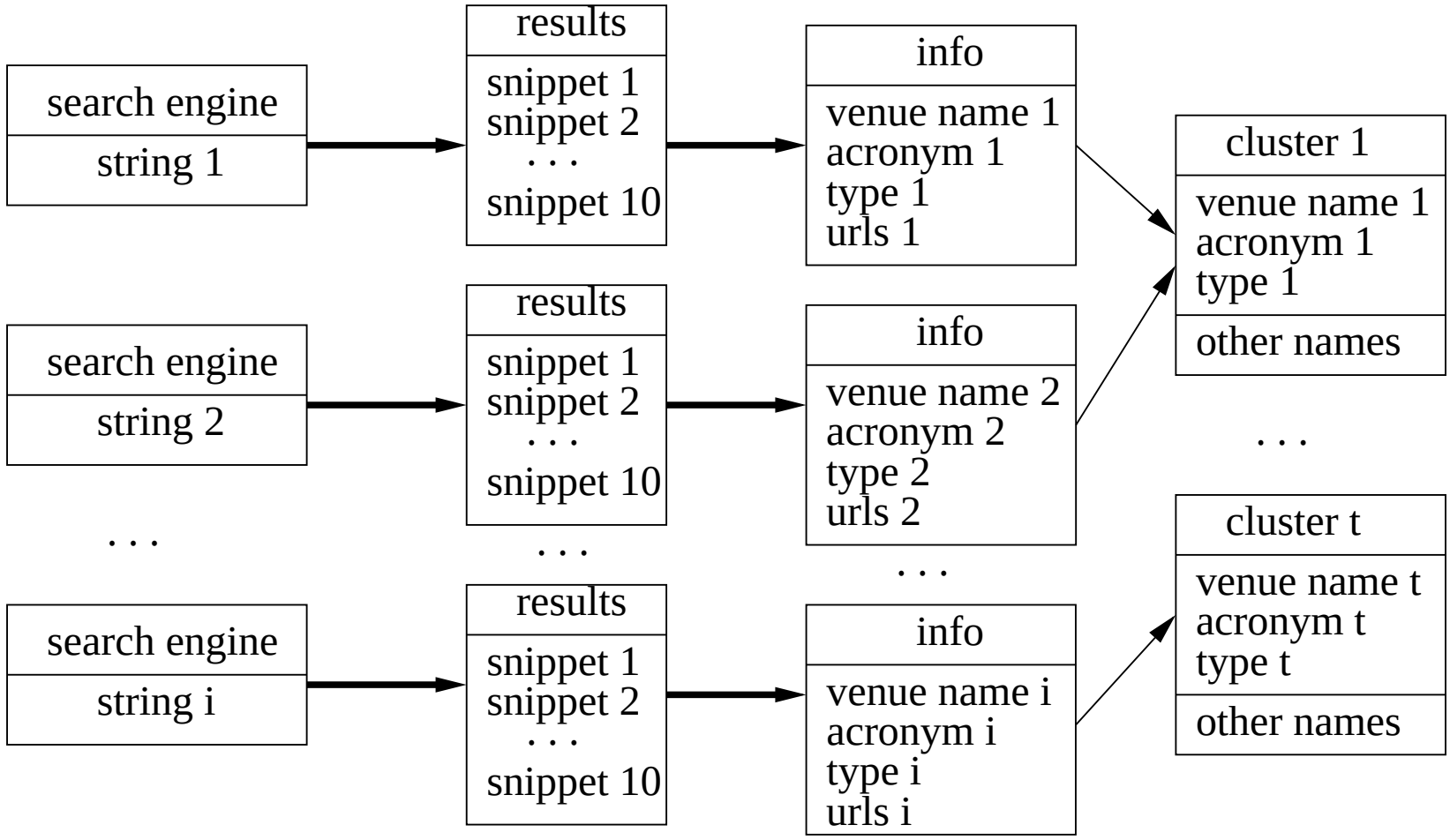


Step 1

Scheme for Creating PVAF



Scheme for Creating PVAF



Step 1 - Querying the Search Engine

- Normalizing input publication venue strings
  - common abbreviations are expanded
  - stop words are removed
  - additional information is removed
- Submit the query

Step 2 - Extracting Information

- Extracting information from each snippet
  - publication venue name
  - acronym
  - type
  - URL

Publication Venue Name Extraction

- Phrases are extracted from the snippet
- The most similar phrase to the query is selected as the representative from that snippet

Acronym Extraction

- Candidates for acronym are extracted from the snippet
- Candidate acronyms are expanded using the publication venue name selected from the snippet
- The acronym with the highest expansion coefficient is selected to represent the snippet

Venue Type Extraction

- Direct extraction: based on the words that define the publication venue name
- Indirect extraction: based on other pieces of evidence

Step 2 - Extracting Information

- Extracting information from each snippet
  - publication venue name
  - acronym
  - type
  - URL
- Selecting information from the set of snippets
  - publication venue name
  - acronym
  - type
  - set of URLs

Selection of Publication Venue Name

- Compute the sum of similarities among strings

$$sumSim(n_i(t_j)) = \sum_{j \neq k} J(n_i(t_j), n_i(t_k))$$

- Select the publication venue name with the highest sum of similarities

$nMax(s_i) = n_i(t_j)$, such that $n_i(t_j)$ has the maximum $sumSim(n_i(t_j))$

Selection of Publication Venue Acronym

- Select the most frequent acronym from the set of snippets

$aMax(s_i) = a_i(t_j)$, such that $a_i(t_j)$ has the maximum $aCount(a_i(t_j))$

Selection of Publication Venue Type

- Select the most frequent venue type from the set of snippets

$tMax(s_i) = tp_i(t_j)$, such that $tp_i(t_j)$ has the maximum $tCount(tp_i(t_j))$

Step 3 - Clustering the Data

- Cluster the information selected in Step 2
- Generate canonical data

Similarity Function

$$\begin{cases} sim(s_i, s_j) = w_n R_n(s_i, s_j) + w_a R_a(s_i, s_j) + \\ \qquad\qquad w_t R_t(s_i, s_j) + w_u R_u(s_i, s_j) \\ w_n + w_a + w_t + w_u = 1 \end{cases}$$

- $R_n(s_i, s_j)$ is a ranking function that compares the names associated with the venue strings s_i and s_j
- Analogously, $R_a(s_i, s_j)$, $R_t(s_i, s_j)$, and $R_u(s_i, s_j)$ are ranking functions that compare a pair of venue acronyms, a pair of venue types, and a pair of URL sets.
- w_n, w_a, w_t, and w_u are weights used to fine tune the contribution of each of the attributes

Ranking Functions

- Ranking function $R_n(s_i, s_j)$ for publication venue names

$$R_n(s_i, s_j) = \frac{|nMax(s_i) \cap nMax(s_j)|}{|nMax(s_i) \cup nMax(s_j)|}$$

- Ranking function $R_a(s_i, s_j)$ for acronyms

$$R_a(s_i, s_j) = \begin{cases} 1.0 & \text{if } aMax(s_i) = aMax(s_j) \\ & \text{and } aMax(s_i) \text{ is not null} \\ 0.0 & \text{if } aMax(s_i) \neq aMax(s_j) \\ 0.7 & \text{if } aMax(s_i) \text{ is null} \\ & \text{and } aMax(s_j) \text{ is null} \end{cases}$$

Ranking Functions

- Ranking function $R_t(s_i, s_j)$ for types

$$R_t(s_i, s_j) = \begin{cases} 1.0 & \text{if } tMax(s_i) = tMax(s_j) \\ 0.0 & \text{if } tMax(s_i) \neq tMax(s_j) \end{cases}$$

- Ranking function $R_u(s_i, s_j)$ for URLs

$$R_u(s_i, s_j) = \begin{cases} 1.0 & \text{if } |uSet(s_i) \cap uSet(s_j)| \geq 2 \\ 0.5 & \text{if } |uSet(s_i) \cap uSet(s_j)| = 1 \\ 0.0 & \text{if } |uSet(s_i) \cap uSet(s_j)| = 0 \end{cases}$$

Generating Canonical Data

- Publication venue canonical name
  - compute the sum of similarities, weighted by the support, among publication venues in cluster
  - select the publication venue name with the highest sum of similarities
- Publication venue acronym: select the most frequent acronym from the cluster
- Publication venue type: select the most frequent venue type from the cluster

Experimental Evaluation

- Goal: compare the Web-based method with the French, Powell and Schulman's method
- Test Dataset
  - Real dataset of citations from Google Scholar
  - Composed of $16,594$ citation records, with $6,544$ distinct publication venue strings

Sample Test Bases

- *sample-at-random*
  - 100 clusters randomly selected
  - average of 3.5 strings per cluster, the largest cluster has 25 strings, and 46 single clusters
- *sample-of-the-largest*
  - 50 clusters with the largest support
  - average of 11 strings per cluster, the largest cluster has 25 strings, and only one single cluster

Evaluation Metrics

- ACP - Average Cluster Purity
- AVP - Average Publication Venue Purity
- K - geometric mean between ACP and AVP

Results for the FPS Method

Method	*sample-at-random* (%)			*sample-of-the-largest* (%)		
	ACP	AVP	K	ACP	AVP	K
edit distance	90.5	54.0	69.9	87.3	38.1	57.6
Jaccard	93.0	51.9	69.5	88.5	41.6	60.6
edit-distance-jaccard	91.9	55.4	71.4	86.3	46.7	63.5

Results for the Web-based Method

Method	*sample-at-random* (%)			*sample-of-the-largest* (%)		
	ACP	AVP	K	ACP	AVP	K
edit-distance-jaccard	91.9	55.4	71.4	86.3	46.7	63.5
Web-based	88.8	70.8	79.3	86.8	71.3	78.7
Gain of the Web-based	-3.5	27.8	11.1	0.58	52.7	23.9

Comments on the Results

- Main gains come from AVP
  – short and long strings are clustered by our method
  – e.g.: “ACM SOSP" and “Symposium on Operating Systems Principles"
- The Web-based method works better with the *sample-of-the-largest* set

Canonical Data

Method	*sample-at-random* (%)			*sample-of-the-largest* (%)		
	name	acron.	type	name	acron.	type
edit-distance-jaccard	55.7	52.3	87.8	57.0	57.5	82.1
Web-based	74.7	71.6	92.6	80.8	80.0	94.0
Gain of the Web-based	34.1	36.9	5.5	41.7	39.1	14.5

Concluding

- Novel method to create publication venue authority files
  - uses the Web to expand citation strings
  - takes advantage of sophisticated matching procedures and of large repositories of Web search engines
- Results indicate large gains in the quality of the authority file produced

Future Work

- To determine how large a publication venue authority file should be to cover a specific area of knowledge, such as Computer Science
- To test our method in other areas of knowledge such as physics, biology, and medical sciences

Questions?